# Climbing the Leaderboard ☆



Alice is playing an arcade game and wants to climb to the top of the leaderboard. Can you help her track her ranking as she plays? The game uses Dense Ranking, so its leaderboard works like this:

- The player with the highest score is ranked number $1$ on the leaderboard.
- Players who have equal scores receive the same ranking number, and the next player(s) receive the immediately following ranking number.

We want to determine Alice's rank as she progresses up the leaderboard. For example, the four players on the leaderboard have high scores of $100$, $90$, $90$, and $80$. Those players will have ranks $1$, $2$, $2$, and $3$, respectively. If Alice's scores are $70$, $80$ and $105$, her rankings after each game are $4^{th}$, $3^{rd}$ and $1^{st}$.

You are given an array, $scores$, of *monotonically decreasing* leaderboard scores, and another array, $alice$, of Alice's scores for the game. You must print $m$ integers. The $j^{th}$ integer should indicate the current rank of alice after her $j^{th}$ game.

**Input Format**

The first line contains an integer $n$, the number of players on the leaderboard.
The next line contains $n$ space-separated integers $scores[i]$, the leaderboard scores in decreasing order.
The next line contains an integer, $m$, denoting the number games Alice plays.
The last line contains $m$ space-separated integers $alice[j]$, her game scores.

**Constraints**

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq m \leq 2 \times 10^5$
- $0 \leq scores[i] \leq 10^9$ for $0 \leq i < n$
- $0 \leq alice[j] \leq 10^9$ for $0 \leq j < m$
- The existing leaderboard, $scores$, is in *descending* order.
- Alice's scores $alice$, are in *ascending* order.

**Subtask**

For $60\%$ of the maximum score:

- $1 \leq n \leq 200$
- $1 \leq m \leq 200$

**Output Format**

Print $m$ integers. The $j^{th}$ integer should indicate the rank of alice after playing the $j^{th}$ game.

**Sample Input 0**

Author Shafaet
Difficulty Medium
Max Score 20
Submitted By 25592

NEED HELP?

View discussions
View editorial
View top submissions

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

Download problem statement
Download sample test cases
Suggest Edits

```
7
100 100 50 40 40 20 10
4
5 25 50 120
```

**Sample Output 0**

```
6
4
2
1
```

**Explanation 0**

Alice starts playing with 7 players already on the leaderboard, which looks like this:

| Rank | Name | Score |
|---|---|---|
| 1 | Emma | 100 |
| 1 | David | 100 |
| 2 | Caroline | 50 |
| 3 | Ritika | 40 |
| 3 | Tom | 40 |
| 4 | Heraldo | 20 |
| 5 | Riley | 10 |

After Alice finishes game 0, her score is 5 and her ranking is 6:

| Rank | Name | Score |
|---|---|---|
| 1 | Emma | 100 |
| 1 | David | 100 |
| 2 | Caroline | 50 |
| 3 | Ritika | 40 |
| 3 | Tom | 40 |
| 4 | Heraldo | 20 |
| 5 | Riley | 10 |
| **6** | **Alice** | **5** |

After Alice finishes game 1, her score is 25 and her ranking is 4:

| Rank | Name | Score |
|---|---|---|
| 1 | Emma | 100 |
| 1 | David | 100 |
| 2 | Caroline | 50 |
| 3 | Ritika | 40 |
| 3 | Tom | 40 |
| **4** | **Alice** | **25** |
| 5 | Heraldo | 20 |
| 6 | Riley | 10 |

After Alice finishes game 2, her score is 50 and her ranking is tied with Caroline at 2:

| Rank | Name | Score |
|---|---|---|
| 1 | Emma | 100 |
| 1 | David | 100 |
| 2 | Caroline | 50 |
| **2** | **Alice** | **50** |
| 3 | Ritika | 40 |
| 3 | Tom | 40 |
| 4 | Heraldo | 20 |
| 5 | Riley | 10 |

After Alice finishes game **3**, her score is **120** and her ranking is **1**:

| Rank | Name | Score |
|---|---|---|
| **1** | **Alice** | **120** |
| 2 | Emma | 100 |
| 2 | David | 100 |
| 3 | Caroline | 50 |
| 4 | Ritika | 40 |
| 4 | Tom | 40 |
| 5 | Heraldo | 20 |
| 6 | Riley | 10 |

**Current Buffer** (saved locally, editable) Java 7

```
10 import java.io.*;
11 import java.math.*;
12 import java.text.*;
13 import java.util.*;
14 import java.util.regex.*;
15
16 public class Solution {
17
18     /*
19      * Complete the climbingLeaderboard function below.
20      */
21     static int[] climbingLeaderboard(int[] scores, int[] alice) {
22         /*
23          * Write your code here.
24          */
25
26     }
27
28     private static final Scanner scanner = new Scanner(System.in);
29
30     public static void main(String[] args) throws IOException {
31         BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));
32
33         int scoresCount = Integer.parseInt(scanner.nextLine().trim());
34
35         int[] scores = new int[scoresCount];
36
37         String[] scoresItems = scanner.nextLine().split(" ");
38
39         for (int scoresItr = 0; scoresItr < scoresCount; scoresItr++) {
40             int scoresItem =
Integer.parseInt(scoresItems[scoresItr].trim());
41             scores[scoresItr] = scoresItem;
42         }
```

```
43
44          int aliceCount = Integer.parseInt(scanner.nextLine().trim());
45
46          int[] alice = new int[aliceCount];
47
48          String[] aliceItems = scanner.nextLine().split(" ");
49
50          for (int aliceItr = 0; aliceItr < aliceCount; aliceItr++) {
51              int aliceItem =
Integer.parseInt(aliceItems[aliceItr].trim());
52              alice[aliceItr] = aliceItem;
53          }
54
55          int[] result = climbingLeaderboard(scores, alice);
56
57          for (int resultItr = 0; resultItr < result.length; resultItr++)
{
58              bufferedWriter.write(String.valueOf(result[resultItr]));
59
60              if (resultItr != result.length - 1) {
61                  bufferedWriter.write("\n");
62              }
63          }
64
65          bufferedWriter.newLine();
66
67          bufferedWriter.close();
68      }
69  }
70
```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code